Project Account: 16-40-201
Project Title: Intelligent Text Retrieval and Knowledge Acquisition from Texts for NASA Applications: Preprocessing Issues
Agency: NASA
Deliverable: Final Report

Abstract

A system that retrieves problem reports from a NASA database is described[1]. The database is queried with natural language questions. Part-of-speech tags are first assigned to each word in the question using a rule-based tagger. A partial parse of the question is then produced with independent sets of deterministic finite state automata. Using partial parse information, a look up strategy searches the database for problem reports relevant to the question. A bigram stemmer and irregular verb conjugates have been incorporated into the system to improve accuracy. The system is evaluated by a set of fifty five questions posed by NASA engineers. A discussion of future research is also presented.

1 Introduction

At the National Aeronautics & Space Administration(NASA), reports documenting technical problems that have been encountered over the years are stored in a continually growing database. These Problem Reports(PRs) can be divided into specific areas such as Mechanical, Fuel Cell, Orbiter Structure, Orbiter Electrical... They are written by NASA engineers at remote terminals and range in size from as few as fifty words to almost two thousand words. This research implements a system which searches this database and returns relevant problem reports to questions which are asked in natural language by NASA engineers.

The system described here uses Brill's tagger(Brill, 1992) to first assign part-of-speech tags to each word in the question. The tagger was not trained on the database of NASA problem reports. To avoid simple errors that occur due to unknown words in the lexicon, a new type of transformation was incorporated into the tagger. Once the text has been tagged, a partial parse of the question is produced with a set of deterministic Finite State Automata(FSAs).

Once a partial parse of the question has been produced, each report in the database is scanned for the noun phrases in the question. A *score* is given to each report based on how many constituents in the noun phrases are matched. If too many reports are given a high score, verb phrase information is used to identify the most relevant reports.

Matching is improved using two techniques 1) a bigram stemmer, and 2) irregular verb conjugates. The primary role of the bigram stemmer is to return reports containing noun phrases with typos, which would otherwise be overlooked. Since the PRs are entered by NASA engineers at remote terminals in an informal manner, many PRs contain typos which can prevent relevant reports from being returned. Irregular verb conjugates are also used to generate more matches in both the initial noun phrase search as well as constrict the number of relevant matches in the secondary verb phrase search. Because of the possibility of nominalization

[1] This research has been supported by NASA Grant 16-40-201.

of irregular verbs, noun phrase searches also benefit by incorporating the conjugate forms of irregular verbs.

The remainder of this paper is as follows: Section 2 describes the partial parsing of the question. Section 3 describes the look up strategy. Section 4 shows results of evaluating the system and Section 5 concludes the paper with a discussion of future research.

2 Interpreting the Question

Part-of-speech(POS) tags[2] are first assigned to each word in the question by Brill's tagger(Brill, 1992). A set of Finite State Automata(FSAs) then identify the syntactic relations in the question. Figure 1 illustrates this process. The rules of each component in this process is read in from separate text files. Rules are in the form of lexical and contextual transformation rule lists for the part-of-speech tagger and FSAs for the other components. Nothing is hardwired in software, the system is entirely declarative.

2.1 The POS Tagger

Because of the time intensive nature of training Brill's tagger, the tagger was used as is. This resulted in some simple errors that would not occur had the tagger been properly trained(Brill, 1995). The main problem is how contextual rules are applied by Brill's tagger. Contextual transformations alter the tagging of a word from X to Y iff either:

1. *The word was not seen in the training corpus OR*
2. *The word was seen tagged Y at least once in the training corpus*

Many words in the problem reports need a part-of-speech tag that did not occur in the Penn Treebank(Marcus et al., 1993) corpus used by Brill to train the tagger. For example, the word *pitted* only occurs as a past participle(VBN) and past tense(VBD) verb in the Penn Treebank. So even when it was used as *the pitted bearing*, it was tagged as a past participle verb when it should have been an adjective(JJ). None of Brill's contextual transformations could prevent this, since neither 1. or 2. are satisfied. To remedy the problem a new type of transformation was incorporated into the system:

X Y FPREVTAG Z

which changes the tag of $word_a$ from *X* to *Y* if it is preceded by $word_b$ which is tagged *Z*, regardless of whether or not $word_a$ includes the *Y* tag in the lexicon. To correctly tag *the pitted bearing*, the transformation was instantiated as *VBN JJ FPREVTAG DT* – where DT is the tag for determiner. Another instantiation was included to change a word that is tagged VBN to JJ when it is preceded by a preposition or subordinating conjunction: *VBN JJ FPREVTAG IN*, as in *with pitted bearings*. Adding these transformations corrected many simple errors without producing any of their own.

2.2 The Finite State Approach

Our approach to partial parsing is opposite to Abney's (1996) *easy-first parsing*. We identify larger relations first and then focus on the smaller constituents comprising them. Consider the example: *John went to the black board after the teacher threatened to expel him.* In the easy-first approach, simple FSAs are first used to capture smaller relations. Noun phrases are first identified, followed by prepositional phrases, subordinate clauses, etc... This would result in *after the teacher* incorrectly being identified as a prepositional phrase.

In our *larger-first parsing* approach, we first use simple FSAs to identify the larger syntactic relations and then identify the smaller constituents inside these relations.

[2] Refer to Santorini (1995) for a thorough description of the Penn Treebank Tagset used by Brill's tagger.

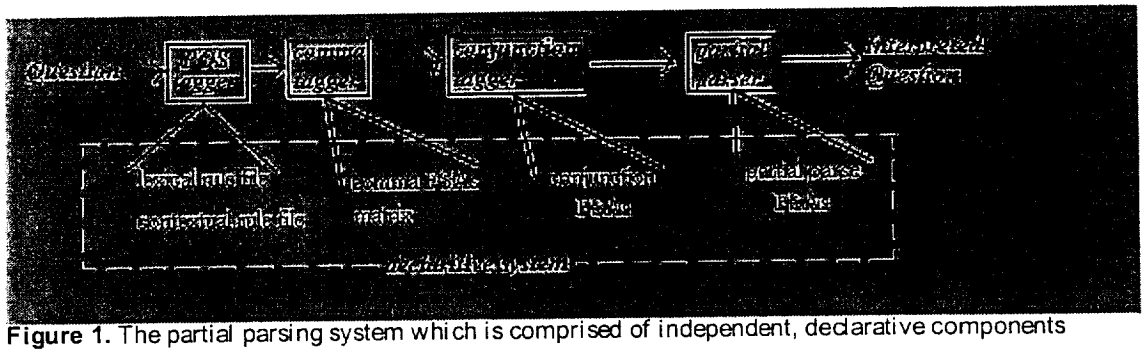

Figure 1. The partial parsing system which is comprised of independent, declarative components

In the example above, the subordinate clause *after the teacher threatened him* would be identified first, followed by the smaller constituents inside of it. Since the Penn Treebank Tagset does not distinguish between a subordinating conjunction and preposition, this error will occur frequently.

A shortcoming of the part-of-speech tagger is that no attempt is made to categorize commas or coordinating conjunctions. Two intermediate components have been added to the system which accomplishes this by assigning descriptive tags to these elements.

Following our larger-first philosophy, the comma tagger(van Delden and Gomez, 2002) is the next component after tagging. Larger relations delimited by commas such as relative clauses, subordinate clauses, and appositions are identified first, as well as commas coordinating a series or pair of syntactic relations. A total of sixteen types of descriptive comma tags have been defined.

An important aspect of the comma is that it can delimit numerous syntactic relations simultaneously. For example, *In the Fall of 1992, a great year for sports, my favorite team won the World Series.* Here the first comma concludes a prepositional phrase, but also introduces an apposition. Every comma FSA is considered for every comma in the sentence, and a temporary set of comma tags are assigned to each comma. A co-occurrence matrix identifies which comma tags can occur with each other, removing incorrect co-occurrences. This is different to the conjunction and partial parsing application of FSAs since conjunctions and syntactic relations require only one descriptive tag. Van Delden and Gomez (2002) also show that this co-occurrence matrix can be automatically acquired using transformation-based learning techniques(Ngai and Florian, 2001 ; Brill, 1995).

Comma tag information is used by the conjunction tagger as well as the partial parser. Since many coordinating conjunctions occur in close proximity to commas, we found that over 30% of conjunctions could automatically acquire their tag from comma tag information. These results were obtained by an inspection of Section 23 of the Wall Street Journal Penn Treebank 3. Since the comma tagger achieves 95% accuracy on correctly tagged text, this proves to be a good bases for the conjunction tagger.

Relying on part-of-speech, comma, and conjunction tags, the partial parser FSAs produces the final partial parse of the sentence. The syntactic relations identified here will be used in the searching strategy.

3 Search Strategy

First the noun phrases which have been identified by the partial parser are searched for in the problem reports. Each report that matches at least one constituent of a noun phrase is given a score and recorded. Pronouns and determiners are removed from the noun phrases. The searching strategy is as follows:

```
report-score = 0
FOR EACH noun phrase in question
  np-score = 0
  FOR EACH sentence in report
    t-score = # of constituents in common
    IF(np-score < t-score)
        np-score = t-score
  report-score = report-score + np-score
```

Each noun phrase is compared with each sentence in the report. The sentence which matches the most constituents in the noun phrase determines the score of that particular noun phrase. The constituents of the noun phrase are matched in the context of the entire sentence disregarding the order of the constituents. All noun phrases in the question generate scores for a particular report. These are combined to form a total *report score*. For example, the following question and example text illustrates this idea:

Is there a large white house in New York for sale?

Mr. York purchased a boat.
A house which is large and white is for sale.
The house is in New York.

Since pronouns and determiners are disregarded, the three noun phrases that will be searched for are: *large white house*, *New York*, and *sale*. *Large white house* will generate a score of 3 for the second sentence. This will remain its score since the third sentence will only generate a score of 1. *New York* will first be given a score of 1 by the first sentence, but the last sentence will change this score to 2. Finally, *sale* generates a score of 1 from the second sentence. The total score would be 6 for this sample text, given the question asked.

Identifying noun phrases and searching for them only in the context of a single sentence reduces the number of irrelevant reports returned. For example, consider the question: *Is there a large white boat for sale in New York?* If a mere match is performed between the noun phrase constituents across any sentence and in any order, this sample text above will also generate a score of 6 for this question. This is undesired since the sample text is clearly more relevant to the first question than the second one. Using our method, the sample text will only generate a score of 5 for the second question, making the sample text less relevant for this question.

Once all reports have been scored. The report with the best score is chosen first and returned to the user. Any remaining reports with a score that at least 75% of the best score are also returned. If there is a tie for the best score, one report is randomly chosen and returned first. If there are more than five reports with a score of at least 75% of the best score, verb phrase information is used to narrow in on the most relevant reports – see Section 3.2.

3.1 Enhancing Noun Phrase Matching

Two enhancements were made to increase the number of relevant reports returned: 1) a bigram stemmer, and 2) irregular verb conjugates.

Instead of only including morphological variations of the constituents in the noun phrase searches, the bigram stemmer(Frakes, 1992) returned more relevant reports because of the nature of the problem reports. Since the problem reports were inputted by engineers at remote terminals in an informal manner, they contained not only several typos but abbreviations that would not be recognized by morphology. For example, trying to match *the broken bearing* with *the brokn bearing*. Neither morphology or irregular verb conjugates with help to match *broken* with *brokn*. This could have been a typo by the NASA engineer or probably just an impromptu abbreviation of the word while entering the report. The bigram stemmer calculates a similarity measure between the two words, in our case Dice's coefficient was used:

$$S = 2C/(A + B)$$

where A is the number of unique bigrams in the first word, B is the number of unique bigrams in the second word, and C is the

number of bigrams shared by A and B. Applying this to our previous example:

broken => br ro ok ke en => A is 5

brokn => br ro ok kn => B is 4

C is 3 – br, ro and ok

This yields a similarity measure of:

*S = 2*3/(5+4) = 6/9 or .67*

Using the recommended cutoff similarity measure of .6(Frakes, 1992) would produce a match on these words even though there is a misspelling in the word.

Even though a bigram stemmer can be used to match many morphological terms, it cannot find matches between irregular verb conjugates. For example, a bigram stemmer will find a match between *reports* and *reported*, however, it would fail at matching *buy* and *bought*. To remedy this problem a list of irregular verbs was extracted from the Collaborative International Dictionary of English (GCIDE, 1999).

Over four hundred irregular verbs were automatically extracted from this resource. This list proved to be useful in matching noun phrases as well as verb phrases, because of the nominalization of irregular verbs. For example, consider the following question that was posed by a NASA engineer: *What causes excessive wear on the ET door latch paddles?* A problem occurs with the noun phrase *excessive wear*. *Wear* is a nominalization of the irregular verb *to wear*. In the most relevant report to this question, however, only the conjugate form *worn* occurs: *...forward and aft et door latchs are worn where roller hits latch lock indicator....* Including the conjugate form *worn* produced a noun phrase match, increasing the relevance of this problem report.

3.2 Verb Phrase Matching

Verb phrase information is used when too many reports have been produced by the noun phrase matcher. All reports with a score that is at least 75% of the best score are used in the search. The algorithm is similar to that of the noun phrase matcher described earlier in Section 3. The verb score of the report is added to the report score that was generated by the noun phrase matcher. Once all reports have been evaluated, the report with the new best score is chosen and returned along with any other reports which have a score of at least 75% of the best score.

If there are still many reports with scores over 75% of the best score, only the top five are returned. In cases such as these, the user is recommended to try re-phrasing the question or making it more specific.

Similar to the noun phrase matcher, both the bigram stemmer and irregular verb conjugates have been incorporated into the verb phrase matcher to improve performance. Auxiliary and modal verbs were removed from verb phrase searches.

4 Evaluation

The system was evaluated with a set of fifty five questions created by NASA engineers. The fifty five questions were asked to the Mechanical database which contains over five hundred problem reports ranging in size from fifty words to almost two thousand words. The results are shown in Table 1. The most relevant report was always found within the top three reports returned, and was the first report found for 82% of the questions.

Table 1. Evaluation results from a set of fifty five test questions created by NASA engineers.

Most Relevant Report Returned	
First	82%
In Top Two	95%
In Top Three	100%

In the case of a tie of the best report scores, the best reports are returned in random order. For example, when the most relevant report was returned second, it sometimes had

a score equal to the first report returned. Similarly, when the most relevant report was returned first, the second report sometimes had an equally good score. The aim of this research was not to always return the most relevant report first, but within the top five. The strategy described here exceeded our expectations for the set of test questions by returning the most relevant report within the top three. Since there were about five hundred reports, this can be considered a filtering of irrelevant reports with an accuracy of 99.4%.

5 Future Research

This system so far solves an *information retrieval* problem by incorporating natural language processing. A small set of relevant reports are filtered out of a larger database. Currently, the database has about five hundred reports and the system returns the most relevant report within the top three. A NASA engineer can use this system to find previous dispositions to problems that have already been encountered.

Future research on this project is two fold: 1) Enhancing information retrieval, and 2) Extracting knowledge from the most relevant reports.

As more problem reports are added to the database, the information retrieval techniques current employed maybe not be sufficient to always return the most relevant problem reports. Since our partial parsing system assigns syntactic roles to the commas in the question, the search strategy could be enhanced to incorporate this information. For example, consider the test question: *Has pitting increased on MWA inner bearing races, which were flown with known pitting?* Using comma information, *which were flown with known pitting* is identified as a relative clause. When searching for the noun phrase *MWA inner bearing races*, a higher score can be given to the report in which *MWA inner bearing races* is being modified by *flown with known pitting* or a variation there of. Currently, even if these two relations occur in separate sentences, they will receive the same score as when they are found in the same sentence. Similarly, the conjunction tags may be incorporated in the search strategy to score reports differently.

WordNet(Miller, 1993) might also enhance the search strategy. Incorporating WordNet synsets has been shown to improve recall (Smeaton et al., 1995) because more relevant documents, that do not include the specific query terms, are matched. However, there have been few successful applications using WordNet to improve precision. Gonzalo et al. (1998), however, does describe how incorporating WordNet synsets in the indexing space improve performance. Including some of these techniques may also improve the accuracy of our system.

The second focus of future research is to extract knowledge from the most relevant reports that have been filtered out of the database. After a partial parse of both the question and the most relevant problem reports are created, a more sophisticated analysis can be performed to directly answer the question. For example, return a *yes* or *no* for a yes/no question, including an explanation which supports your answer.

The first step in this process would be a semantic analysis of the sentence. Gomez (2001, 1998) explains an algorithm which uses an enhanced WordNet to determine the meaning of the verb as well as its thematic roles, adjuncts and prepositional phrase attachment. The algorithm is based on predicates that have been defined by Gomez for WordNet verb classes. The thematic roles in the predicates contain information about the grammatical relations and ontological categories that realize them. The ontological categories belong to WordNet noun ontology, which has undergone reorganization and modification as dictated by the semantic interpretation algorithm. The use of the semantic interpreter will allow not only to obtain more precise answers, but also to mine the texts in search of semantic patterns across a large number of reports.

References

Steven Abney. 1996. *Partial Parsing via Finite-State Cascades.* In Proceedings of the ESSLLI'96 Robust Parsing Workshop.

Eric Brill. 1995. *Transformation-based error-driven learning and natural language processing: A case study in part of speech tagging* Computational Linguistics, 21(4):543-565.

Eric Brill. 1992. *A Simple Rule-Based Part of Speech Tagger.* In the Third Conference on Applied Natural Language Processing, Pages 152-155, Trento, Italy.

W. Frakes. *Stemming algorithms.* Information Retrieval: Data Structures & Algorithms, pages 131--160. Prentice Hall, 1992.

M. Marcus, B. Santorini, and M. Marcinkiewiecz. 1993. *Building a large annotated corpus for English.* Computational Linguistics, 19 (2):313-330.

G. Ngai and R. Florian. 2001. *Transformation-Based Learning in the Fast Lane.* North American Chapter of the Association for Computational Linguistics.

GCIDE 1999. *The Collaborative International Dictionary of English – GCIDE.* **Version 0.41. February, 1999.**

George Miller. 1993. *Introduction to WordNet: An On-line Lexical Database*, Princeton, CSL Report 43.

Gonzalo, J., F. Verdejo, I. Chugur and J. CigarrÃ¡. 1998. *Indexing with WordNet synsets can improve text retrieval,* in ACL/COLING Workshop on Usage of WordNet for Natural Language Processing.

Fernando Gomez. 2001. *An Algorithm for Aspects of Semantic Interpretation Using an Enhanced WordNet.* North American Chapter of the Association of Computational Linguistics, NAA CL2001

Fernando Gomez. 1998. *Linking WordNet Verb Classes to Semantic Interpretation.* Proceedings of the COLING-ACL Workshop on the Usage of WordNet on NLP Systems. Universite de Montreal, Quebec, Canada, Agust, 16 98 pp. 58-64.

Peter Ruber. 2001. *Asking Naturally: Vendors of natural language query tools hope to attract enterprise customers.* Knowledge Management. August 2001

Beatrice Santorini. 1995. *Part-of-speech Tagging Guidelines for the Penn Treebank Project,* 3rd Revision, 2nd Printing.

A. Smeaton, F. Kelly, and R O'Donnell. 1995. *TREC-4 Experiments at Dublin City University: Thresholding posting lists, query expension with WordNet and POS tagging of Spanish.* In Proceedings of TREC-4.

Sebastian van Delden and Fernando Gomez. 2002. *Combining Finite State Automata and Transformation-based Learning to Determine the Syntactic Roles of Commas.* University of Central Florida. Technical Report TR-CS-02-01.